Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

High Performance Computing with GAMS

F. Fiand, M. Bussieck

September 7th, 2017

GAMS Software GmbH

A PROJECT BY

GAMS · TU Berlin · HLRS · Jülich Forschungszentrum · DLR Deutsches Zentrum für Luft- und Raumfahrt German Aerospace Center · ZIB

Outline

- Introduction
- Model Annotation
- Distributed Model Generation
- Outlook

Introduction

GAMS – System Overview

Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building:

Model is executable algebraic description of optimization problem.

GAMS – System Overview

BEAM-ME

Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building:

Model is executable algebraic description of optimization problem.

$$\sum_{p \in P: rp_{r,p}} \mathrm{POWER}_{t,r,p} + \sum_{r2 \in R: net_{r2,r}} (\mathrm{FLOW}_{t,r2,r}) - \sum_{r2: net_{r,r2}} \mathrm{FLOW}_{t,r,r2} + \sum_{s \in S: rs_{r,s}} (\mathrm{STORAGE\_OUTFLOW}_{t,r,s} - \mathrm{STORAGE\_INFLOW}_{t,r,s}) \geq \mathrm{demand}_{t,r} \quad \forall t \in T, r \in R$$

```
eq_power_balance(t,r)..
  sum(rp(r,p),    POWER(t,r,p))
+ sum(net(r2,r), FLOW(t,net)) - sum(net(r,r2), FLOW(t,net))
+ sum(rs(r,s),    STORAGE_OUTFLOW(t,r,s) - STORAGE_INFLOW(t,r,s)) =g= demand(t,r);
```

Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building

Declarative elements

- **Similar to mathematical notation**
- **Easy to learn - few basic language elements: sets, parameters, variables, equations, models**
- **Model is executable (algebraic) description of the problem**

GAMS – System Overview

BEAM-ME

Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building

Declarative elements

- Similar to mathematical notation
- Easy to learn - few basic language elements: sets, parameters, variables, equations, models
- Model is executable (algebraic) description of the problem

Procedural elements

- Control Flow Statements (e.g. loops, for, if,...),
- Build complex problem algorithms within GAMS
- Simplified interaction with other systems
  - Data exchange
  - GAMS process control

GAMS – System Overview

BEAM-ME

- Broad Range of application areas

Agricultural Economics	Applied General Equilibrium
Chemical Engineering	Economic Development
Econometrics	Energy
Environmental Economics	Engineering
Finance	Forestry
International Trade	Logistics
Macro Economics	Military
Management Science/OR	Mathematics
Micro Economics	Physics

GAMS – System Overview

BEAM-ME

- Broad Range of application areas

Agricultural Economics	Applied General Equilibrium
Chemical Engineering	Economic Development
Econometrics	**Energy**
Environmental Economics	Engineering
Finance	Forestry
International Trade	Logistics
Macro Economics	Military
Management Science/OR	Mathematics
Micro Economics	Physics

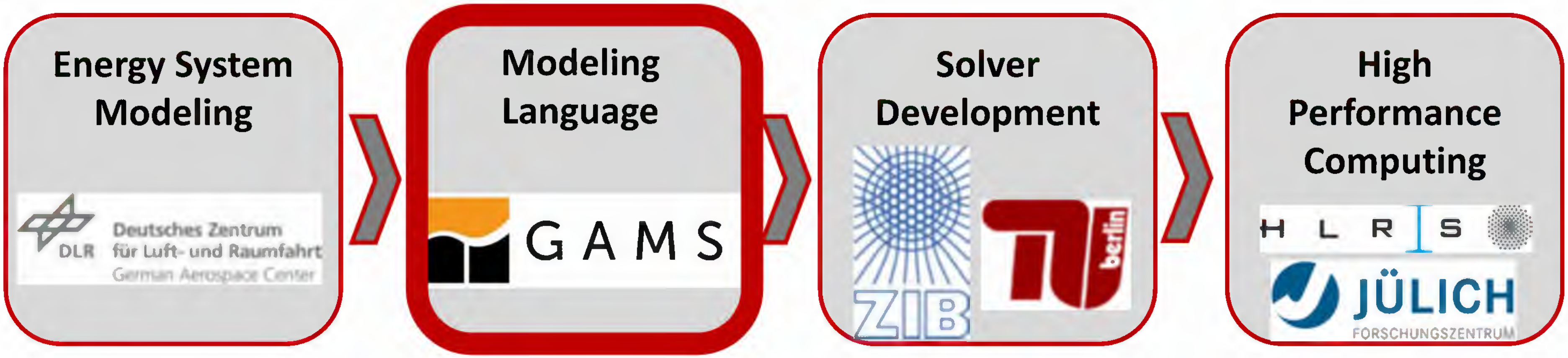
- GAMS is widespread in the ESM community: http://www.energyplan.eu/othertools/

BEAM-ME: An Interdisciplinary Approach

Energy System Modeling → **Modeling Language** → **Solver Development** → **High Performance Computing**

Goal: *Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models*

BEAM-ME: An Interdisciplinary Approach



Goal: *Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models*

Limitations of “standard” Soft- & Hardware

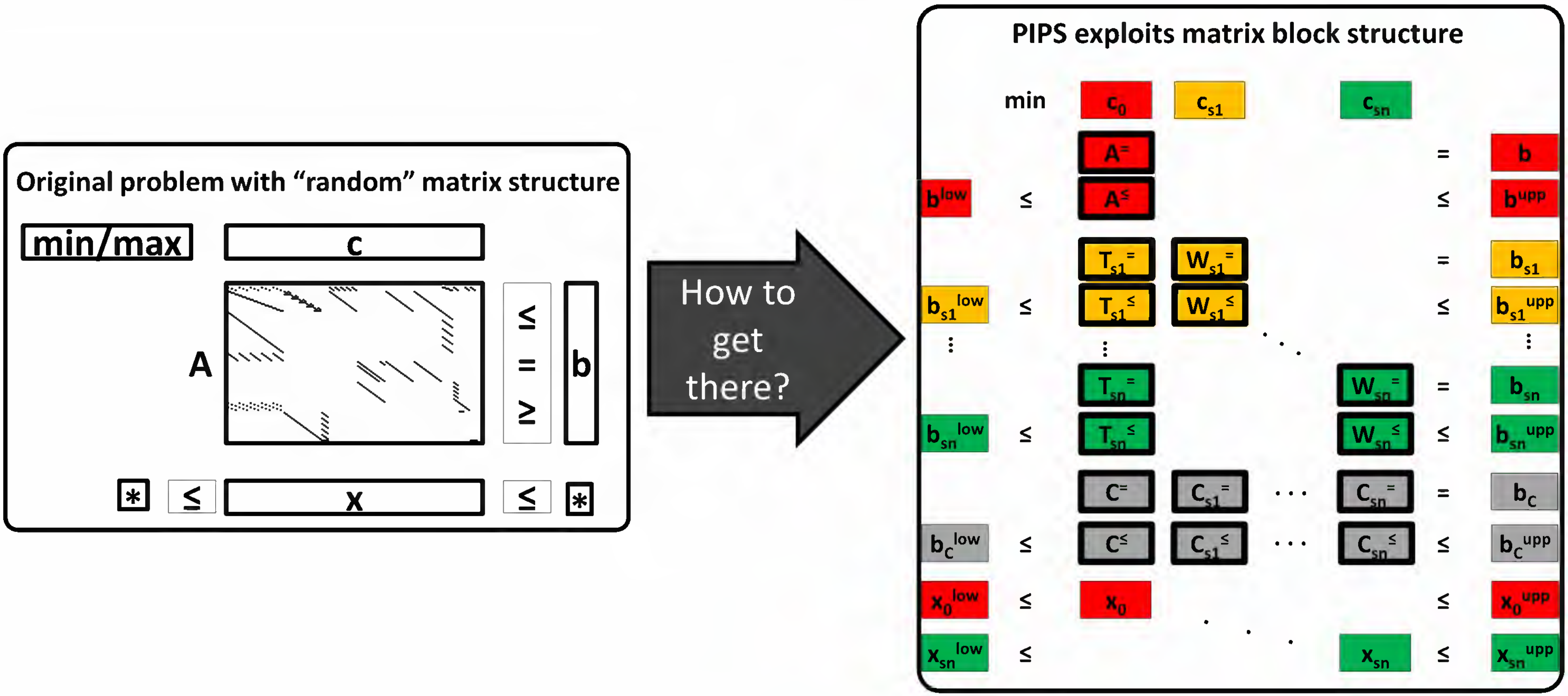
#t	#r	#blocks	#rows (E6)	#cols (E6)	#NZ (E6)	~Mem (GB)	time
730	10	10	0.7	0.8	2.8	2.0	00:01:22
730	10	500	35.0	38.7	142.8	95.7	01:09:36
730	10	2,500	175.3	193.5	713.9	478.8	09:32:55
730	10	4,000	280.5	309.6	1,142.2	767.1	19:22:55
730	10	7,500	526.1	580.5	2,141.2	~1,436.4	-
8,760	10	10	8.4	9.3	34.3	18.2	00:28:57
8,760	10	50	42.1	46.4	171.6	90.4	02:26:25
...							

Test runs were made on JURECA @ JSC

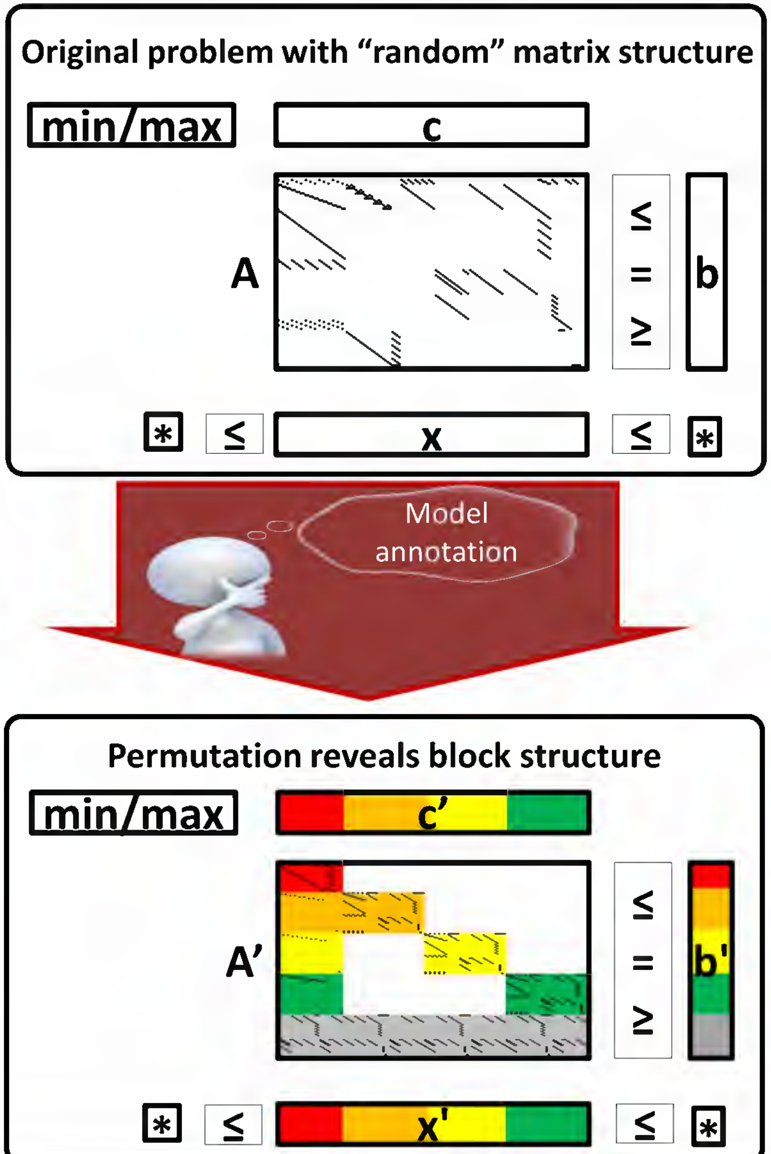
- 2x Intel Xeon E5-2680 v3 (Haswell), 2 x 12 cores @ 2.5GHz
- “fat” node with 1,024 GB Memory
- GAMS 24.8.5 / CPLEX 12.7.1.0
- Barrier Algorithm, Crossover disabled, 24 threads

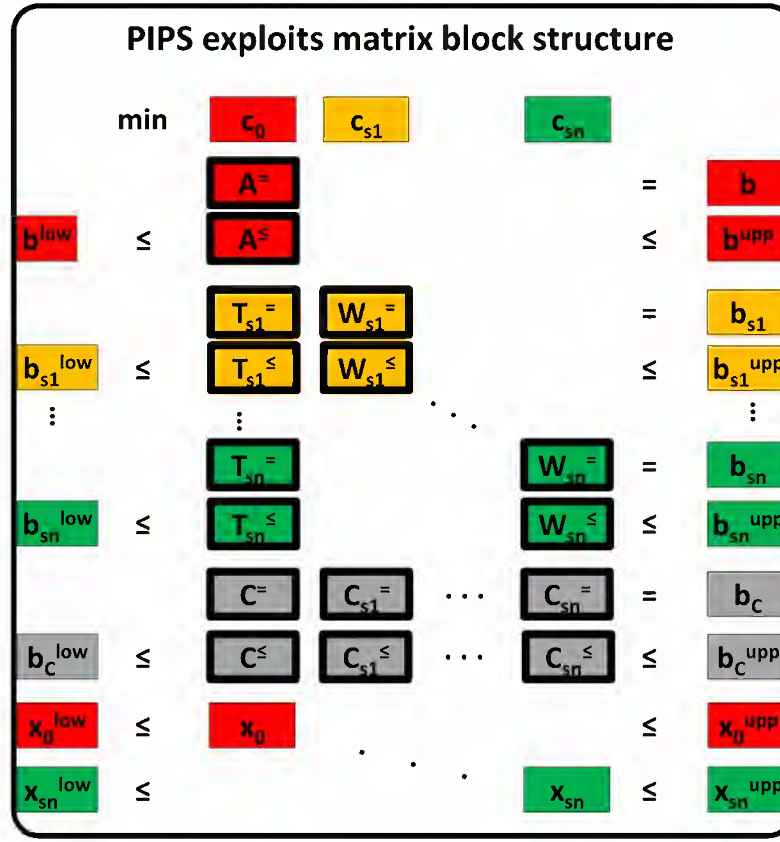
Model Annotation

Model Annotation

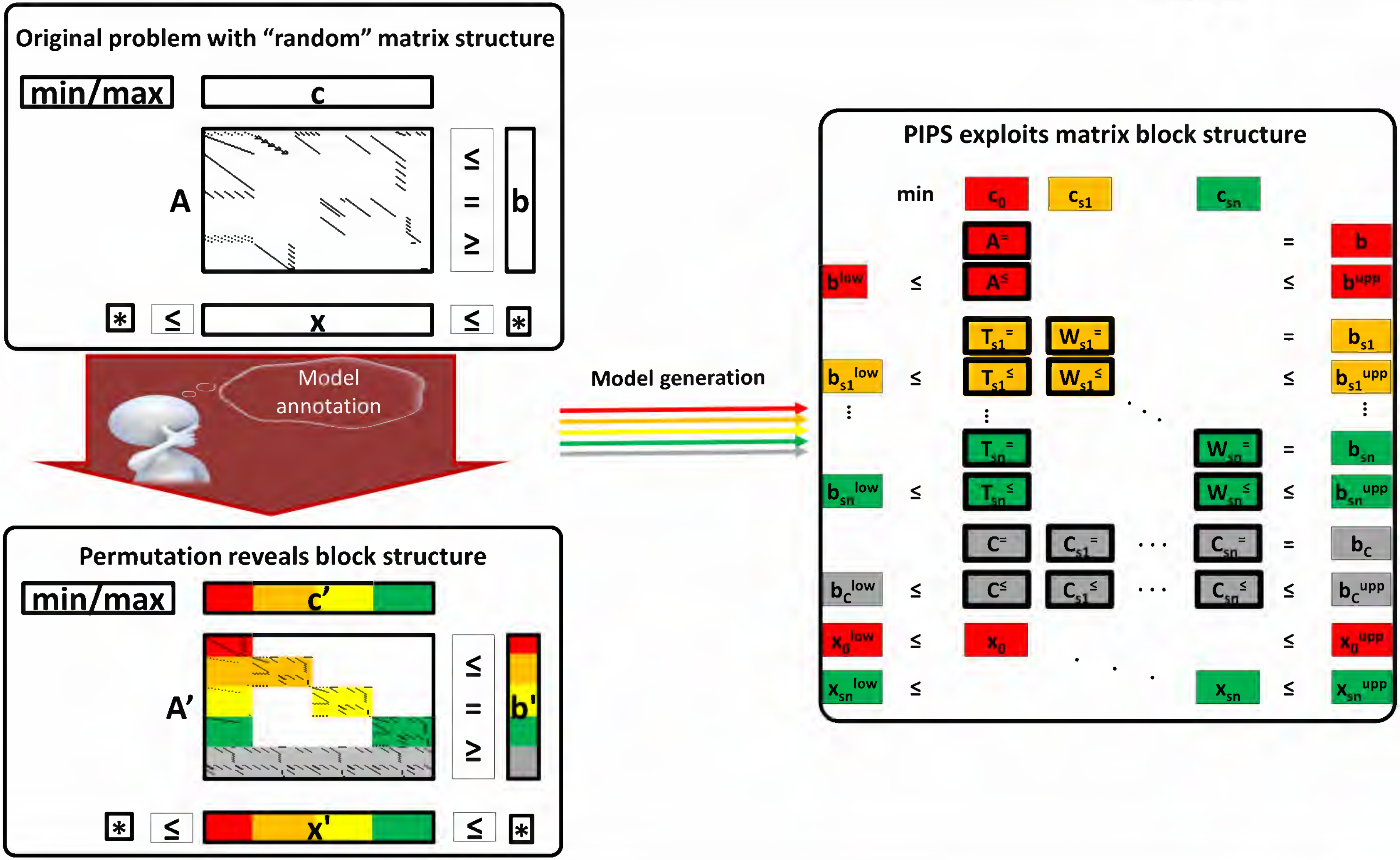


Model Annotation

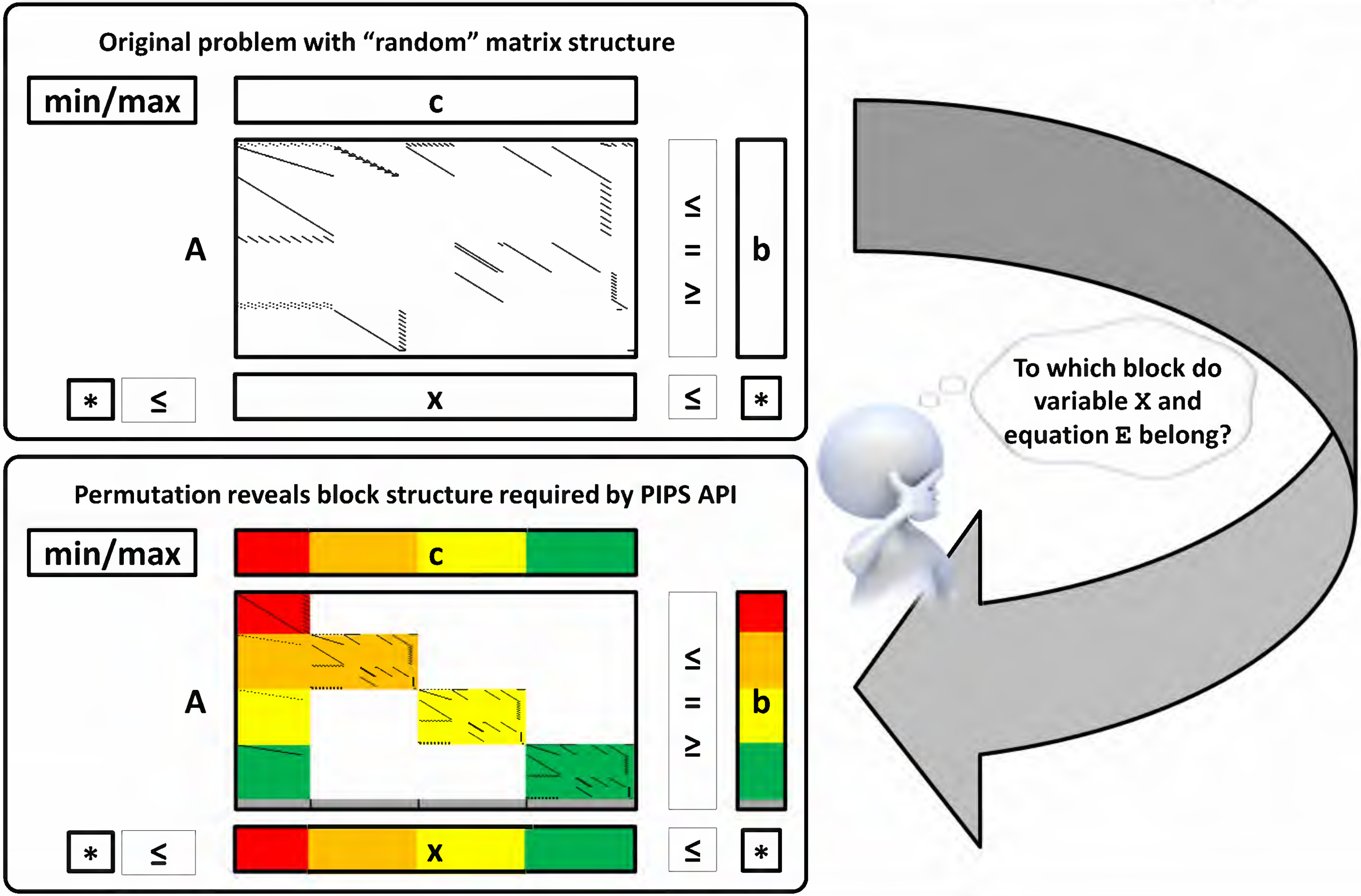




Model Annotation



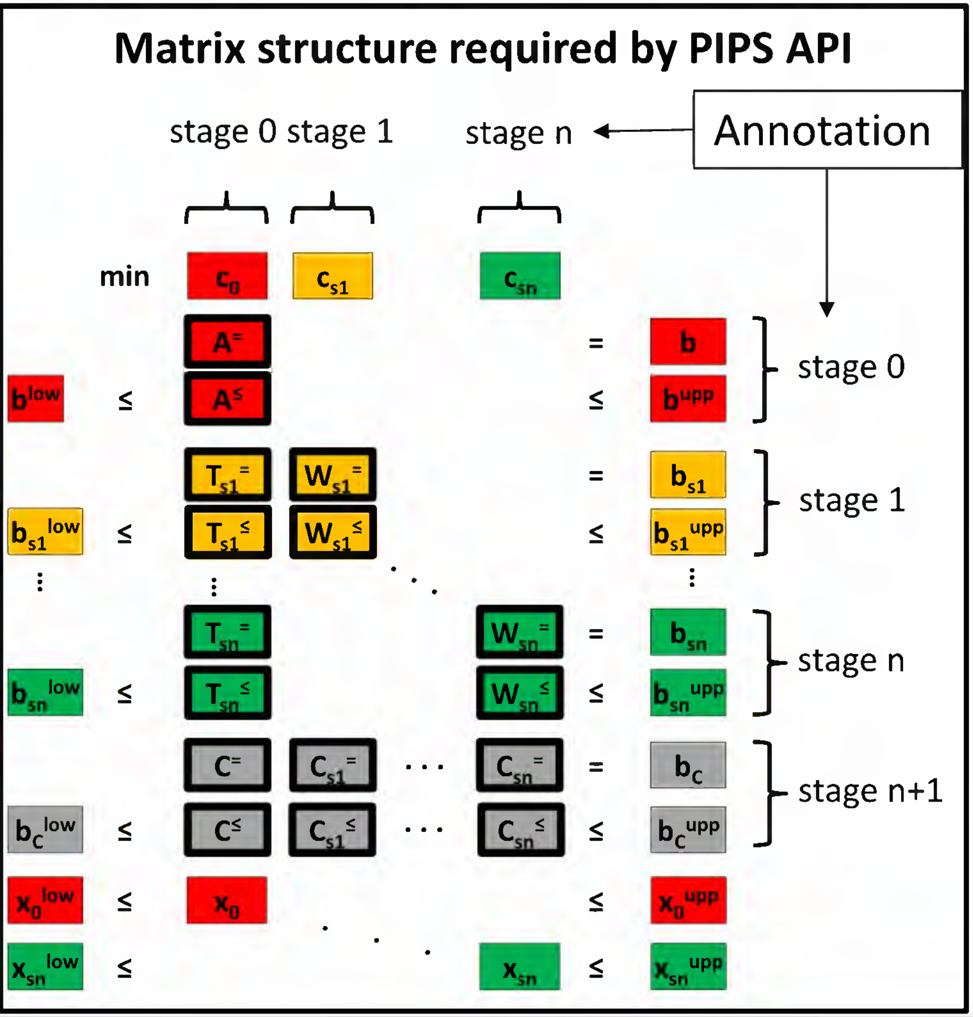
Model Annotation cont.



Model Annotation cont.

Model Annotation by .Stage

- The .stage attribute is available for variables/equations in GAMS

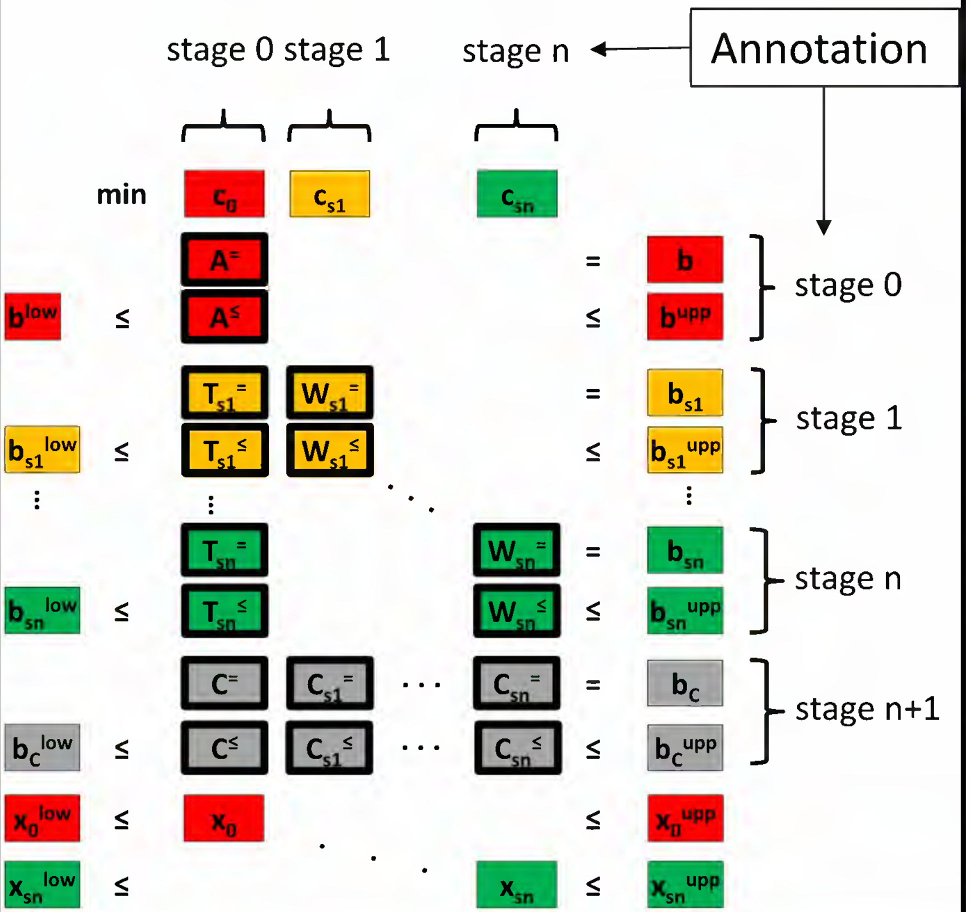


Model Annotation cont.

Model Annotation by .Stage

- The .stage attribute is available for variables/equations in GAMS

Matrix structure required by PIPS API



Exemplary Annotation for SIMPLE model (regional decomposition)

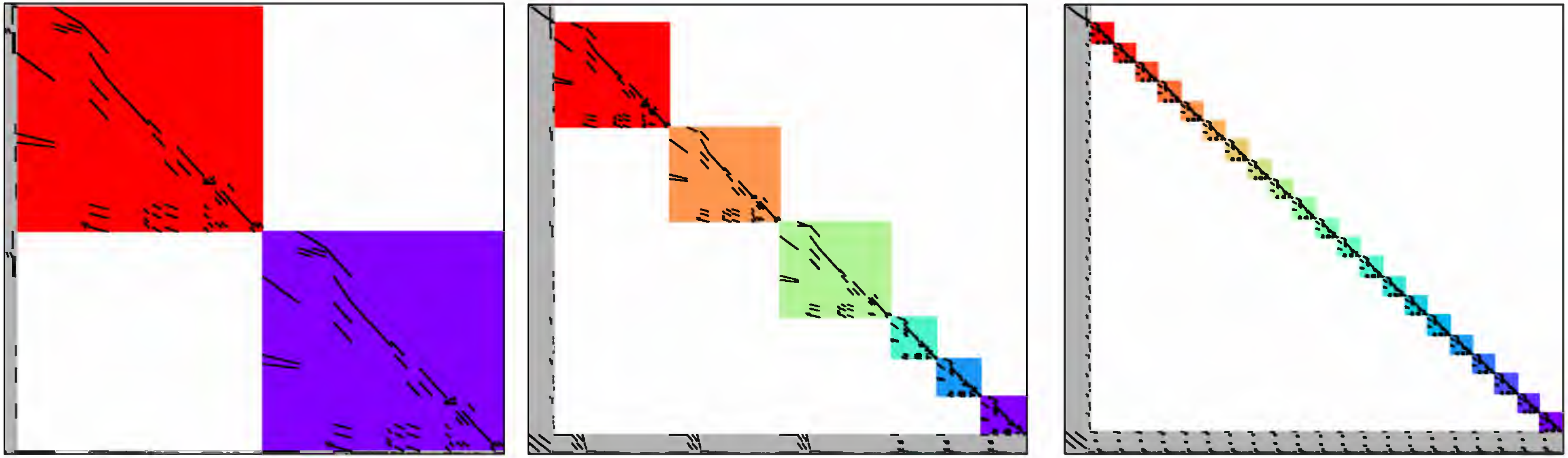
```
Set rr              'regions                  '
    p               'plants                   '
    tt              'time steps               '
    e               'emissions                '
    t(tt)           'subset of active time steps'
    rp(rr,p)        'region to plant mapping  '
    net(rr,rr)      'transmission links       '
;
Alias(rr,rr1,rr2,rr3);
[...]
* Master variables and equation
FLOW.stage(t,net(rr1,rr2))        = 0;
LINK_ADD_CAP.stage(net(rr1,rr2)) = 0;
[...]
* Block variables and equations
POWER.stage(t,rp(rr3,p))      = ord(rr);
EMISSION.stage(rr3,e)         = ord(rr);
[...]
eq_power_balance.stage(t,r)       = ord(rr);
eq_emission_region.stage(rr3,e)  = ord(rr);
[...]
* Linking Equation
eq_emission_cap.stage(e)  = card(rr)+1;
```

Model Annotation cont.

- How to annotate Model depends on how the model should be “decomposed” (by region, time,...)

Model Annotation cont.

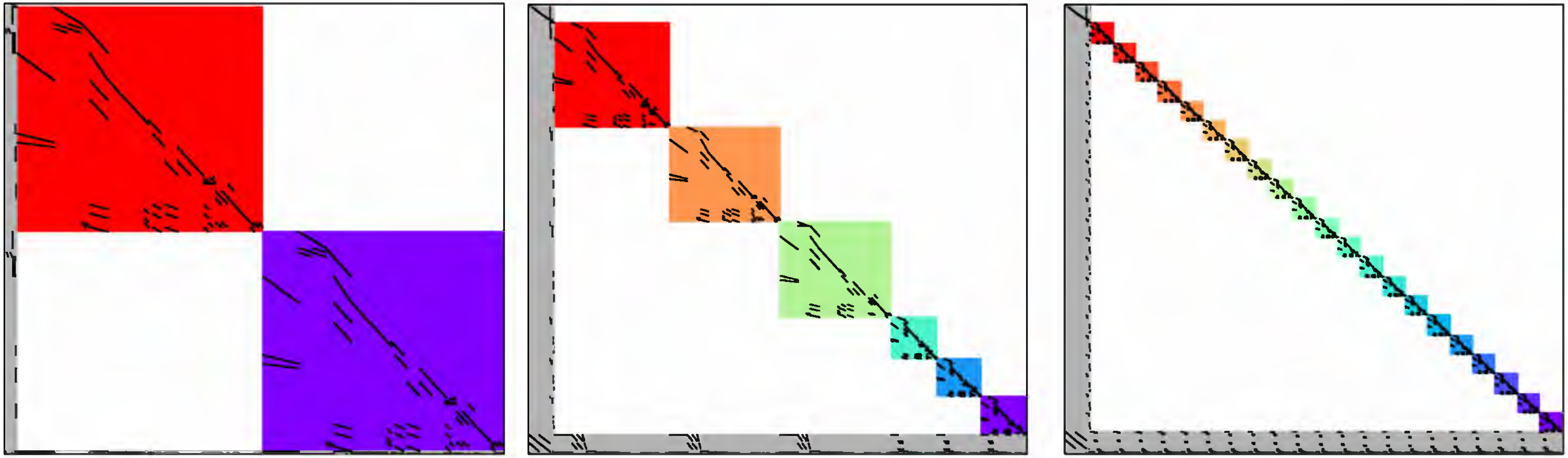
- How to annotate Model depends on how the model should be "decomposed" (by region, time,...)



Plots show three different annotations of identical model

Model Annotation cont.

- How to annotate Model depends on how the model should be "decomposed" (by region, time,...)



Plots show three different annotations of identical model

- How important are blocks of equal size?

Distributed Model Generation

Distributed Model Generation

- "Usual Model": model generation time << solver time
- For LARGE-scale models the model generation may become significant:
  - due to time consumption
  - due to memory consumption
  - due to hard coded limitations of model size (# non-zeroes < ~2.1e9)

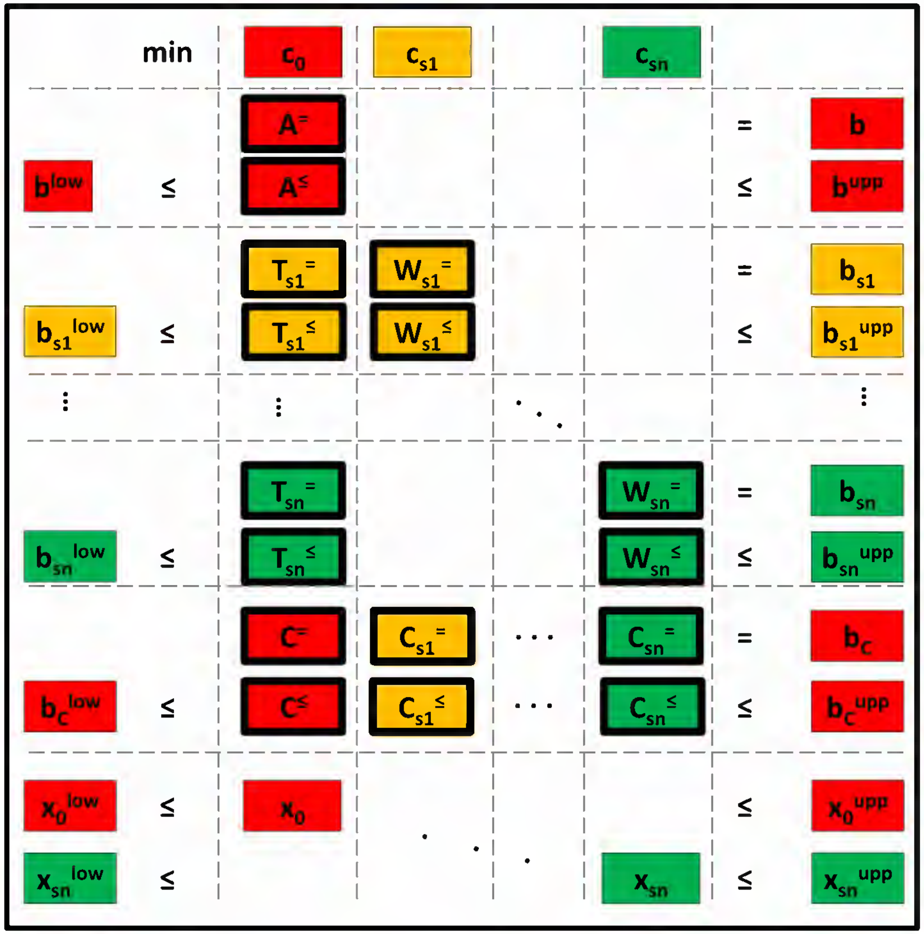
→ Distributed "block-wise" model setup in PIPS-IPM

→ Model annotation determines block membership of all variables and constraints

→ Distributed GAMS processes can generate the separate blocks (model needs to be prepared accordingly!)

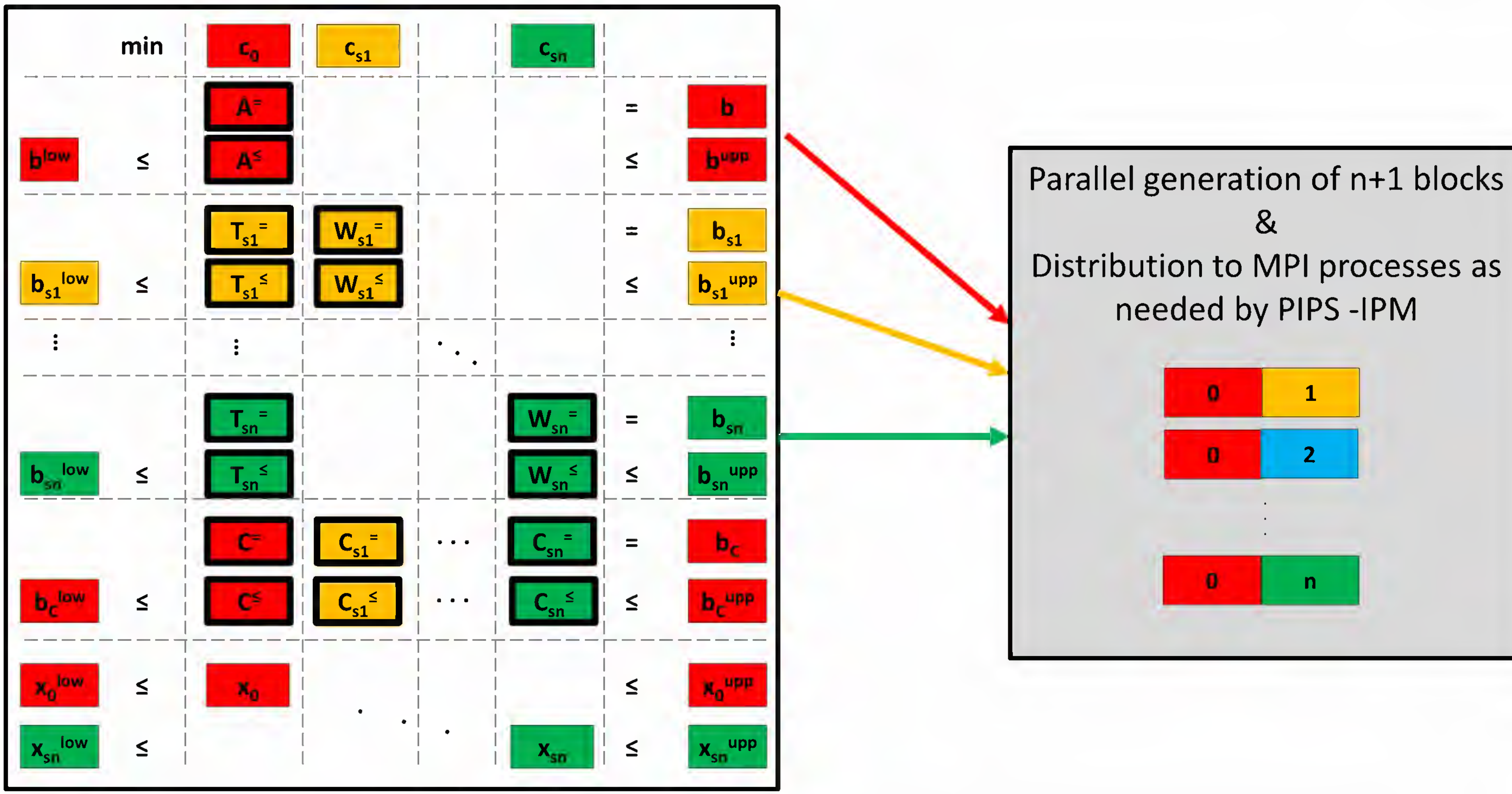
Distributed Model Generation

Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):



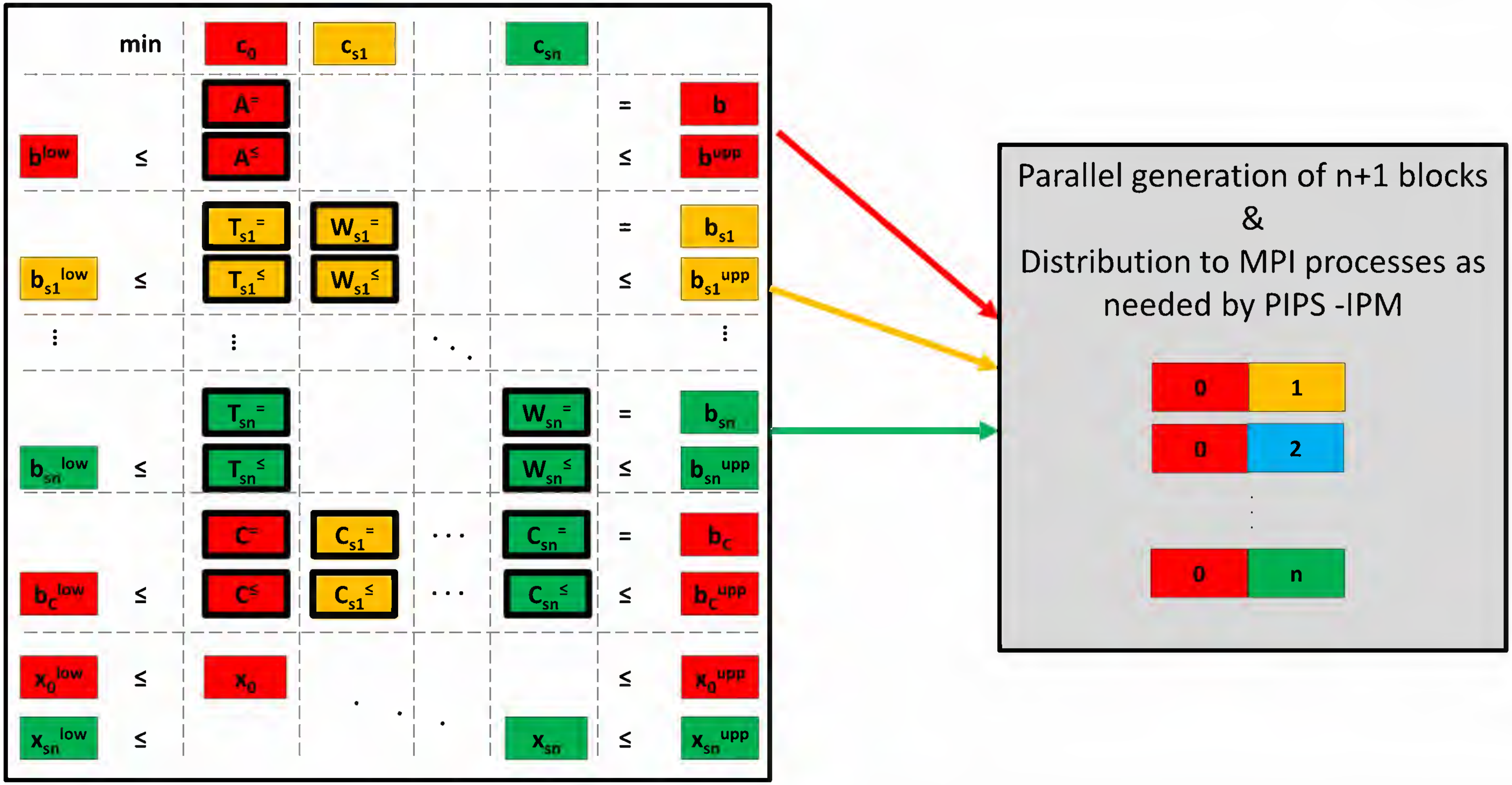
Distributed Model Generation

Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):



Distributed Model Generation

Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):



→ Time to generate n+1 blocks in parallel << time to generate monolithic model

Outlook

Outlook

BEAM-ME

- Model generation and solution are currently separated
  - Integrate those steps into one user friendly process
  - Better user control of GAMS/PIPS
    - options (algorithmic, limits, tolerances)
- Analyze IO bottlenecks (generation/solving)
- Annotation can be adapted for other Decomposition approaches (e.g. CPLEX Benders)
- GAMS-MPI/Embedded Code:
  - Implementation of Benders Decomposition in GAMS for ESM using the GAMS embedded code facility with Python package mpi4py to work with MPI (see talk of L. Westermann, WC-02)

[1] https://www.gams.com/latest/docs/UG_EmbeddedCode.html

Project BEAM-ME

Supported by:

Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

BEAM-ME

A PROJECT BY

GAMS

TU berlin

HLRS

JÜLICH
FORSCHUNGSZENTRUM

DLR
Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

ZIB